



2/6/04

PATENT APPLICATION
ATTORNEY DOCKET NO. 10012379-1

2700
\$

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Kenneth Mark Wilson et al.

Confirmation No.: 7400

Application No.: 09/896,157

Examiner: Gary J. Portka

Filing Date: June 28, 2001

Group Art Unit: 2188

Title: Relocation Table For Use In Memory Management

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

RECEIVED
FEB 11 2004
Technology Center 2100

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith in **triplicate** is the Appeal Brief in this application with respect to the Notice of Appeal filed on Dec. 10, 2003.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) \$330.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

() (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

() one month	\$110.00
() two months	\$420.00
() three months	\$950.00
() four months	\$1480.00

() The extension fee has already been filled in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account **08-2025** the sum of \$330.00. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account **08-2025** pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account **08-2025** under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

(X) I hereby certify that this correspondence is being deposited with the United States Postal Service as Express mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450. Date of Deposit: Feb. 4, 2004 Express Mail Label No.: EL442081774US

() I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number _____ on _____

Number of pages:

Typed Name: Jerome Dechant

Signature: Jerome Dechant

Respectfully submitted,

Kenneth Mark Wilson et al.

By Tuan V. Ngo

Tuan V. Ngo

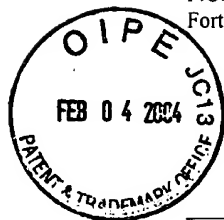
Attorney/Agent for Applicant(s)

Reg. No. **44,259**

Date: **F b. 4, 2004**

Telephone No.: **408 447 8133**

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400



#8
EWM
2-17-04
Docket No.: 10012379-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Kenneth Mark Wilson, et al.

Application No.: 09/896,157

Confirmation No.: 7400

Filed: June 28, 2001

Art Unit: 2188

For: Relocation Table For Use In Memory
Management

Examiner: Gary J. Portka

APPELLANTS' BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RECEIVED

FEB 11 2004

Technology Center 2100

Dear Sir:

This brief is in furtherance of the Notice of Appeal filed in this case on December 10, 2003.

The fees required under § 1.17(f) and any required petition for extension of time for filing this brief and fees therefor are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate.

02/10/2004 DTESSEN1 00000051 082025 09896157

01 FC:1402 330.00 DA

This brief contains items under the following headings as required by 37 C.F.R.
§ 1.192 and M.P.E.P. § 1206:

- I. Real Party In Interest
 - II. Related Appeals and Interferences
 - III. Status of Claims
 - IV. Status of Amendments
 - V. Summary of Invention
 - VI. Issues
 - VII. Grouping of Claims
 - VIII. Arguments
 - IX. Claims Involved in the Appeal
- Appendix A Claims

The final page of this brief bears the attorney's signature.

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is Hewlett-Packard Development Company, L.L.P., a Texas limited liability partnership having its principal place of business in Houston, Texas.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF CLAIMS

- A. Total Number of Claims in the Application: 12 claims, which are identified as claims 1-12.
- B. Current Status of Claims
 - 1. Claims canceled: None
 - 2. Claims withdrawn from consideration but not canceled: None
 - 3. Claims pending: 1-12
 - 4. Claims allowed: None
 - 5. Claims rejected: 1-12
- C. Claims On Appeal: claims 1-12

IV. STATUS OF AMENDMENTS

Appellants filed a Response After Final Rejection on October 16, 2003 without any amendment to the claims. The Examiner responded to the Response After Final Rejection with an Advisory Action mailed November 5, 2003, in which the Examiner stated that Appellants' Reply failed to place the Application in condition for allowance.

V. SUMMARY OF INVENTION

In an embodiment of managing memory in computer systems, each memory page is divided into relocation blocks (220) located at various physical locations, and a relocation table (160) is created with entries (210) used to locate these blocks (220). To access memory for a particular piece of data, a program first uses a virtual address of the data, which, through

a translation look-aside buffer (150), is translated into a physical address within the computer system (600). Using the relocation table (160), the physical address is then translated to a relocation address that identifies the relocation block (220) containing the requested data (Application, Summary, page 4, lines 1-10). If the data intended for the memory access is not in physical memory (120), then the relocation blocks (220) containing the data related to the memory access are loaded into physical memory (120) (FIG 5, step 516; Specification, page 12, lines 10-14).

VI. ISSUE

The issue is whether claims 1-12 are unpatentable under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,297,265 issued to Frank et al. (hereinafter *Frank*).

VII. GROUPING OF CLAIMS

For purposes of this Appeal Brief only, and without conceding the teachings of any prior art reference, the claims have been grouped as indicated below:

Group I	Claims 1- 12
---------	--------------

VIII. ARGUMENTS

Rejection under 35 U.S.C. § 102(b) - Frank

To anticipate a claim under § 102, a reference must teach each and every element of the claim (MPEP § 2131). Appellants respectfully submit that *Frank* fails to teach various elements of pending claims 1-12. Therefore, *Frank* does not anticipate claims 1-12.

1) *Frank* fails to teach, suggest, or make obvious using the cache directories to convert an address of a memory page to an address of a subpage as in Appellants' claim 1 because *Frank's* caches directories do not correspond to Appellants' relocation table

In both the first Office Action and the Final Office Action, the Examiner asserted that “[a]s to claims 1-12, Frank discloses the method, system, and medium as recited, including diving a memory page into a plurality of relocation blocks (“subpages”), and using a relocation table (cache directories) to convert a memory page address to a relocation address.

See Figure 5, and column 11 line 22 through column 12 line 68” (first Office Action, paragraph 3; Final Office Action, paragraph 4).

The Examiner thus corresponded *Frank*’s page to Appellants’ page, *Frank*’s subpages to Appellants’ relocation blocks, and *Frank*’s cache directories to Appellants’ relocation table. Appellants respectfully submit that, if *Frank*’ page corresponds to Appellants’ page and *Frank*’s subpages correspond to Appellants’ relocation blocks, then *Frank* fails to correspond *Frank*’s cache directories to Appellants’ relocation table because *Frank*’s *cache directory* is associated with *pages*, not *subpages* or *relocation blocks*, which have been divided from a *page*. *Frank*’s cache directory records the *association between pages* (*Frank*, col. 12 lines 20-21). In the section Response to Arguments (paragraph 5) of the Final Office Action the Examiner recited *Frank* and admitted that “. . . the *cache directory* records the *association between pages* in the cache and the system *page* . . . ” (emphasis added). *Frank* also discloses that the cache directory acting as a content-address memory allows a cache to *locate a page*: “[e]ach *cache directory* 46A acts as a content-addressable memory. This permits a cache to locate a descriptor for a particular *page* of SVA space . . . ” (col. 12, lines 31-33, emphasis added). Because *Frank*’s cache directories are not associated with subpages, but with *pages*, which do not correspond to Appellants’ *relocation blocks*, *Frank*’s cache directories do not correspond to Appellants’ relocation table. Because *Frank*’s cache directories do not correspond to Appellants’ relocation table, *Frank* cannot logically teach using the *cache directories* to convert an address of the memory page to a relocation address of a relocation block because Appellants’ relocation table in claim 1 is used to convert an address of the memory page to a relocation address of a relocation block.

In paragraph 5 of the Final Office Action, the Examiner explained “[t]he cache may be searched for a corresponding entry for a particular system page (col. 12 lines 31-45), which for each individual cache may be only a subpage of the page, and thus equivalent to a relocation address converted from the system page address using a table (the directories).” Appellants respectfully submit that, even if *Frank*’s cache directories may be used to locate a subpage in the situation where the page has only one subpage, Appellants’ claim 1 is also distinguished from that situation because, in claim 1, “for at least one memory page,” the page has *more than one* relocation block or subpage because that page is divided into a *plurality of relocation blocks*.

2) Frank fails to teach, suggest, or make obvious using the cache directories to locate the relocation blocks at the plurality of locations

As indicated above, because *Frank*'s cache directories are associated with *pages*, *not subpages or relocation blocks* that have been divided from a page, *Frank*'s cache directories do not correspond to Appellants' relocation table. Therefore, without evidence showing otherwise, *Frank*'s cache directories cannot be used to locate the location blocks as in Appellants' claim 1. In the Final Office Action the Examiner indicated that "[t]he cache may be searched for a corresponding entry for a particular system page (col. 12 lines 31-45), which for each individual cache may be only a subpage of the page." Appellants submit that even if the Examiner meant that a cache directory may be used to locate a subpage when the page has only one subpage, claim 1 is also patentably distinguished from that situation because claim 1 recites "for at least one memory page, the page is divided into *a plurality of relocation blocks . . .*" (emphasis added). As a result, at least one page in claim 1 has more than one subpage or relocation block.

On page two of the Advisory Opinion, the Examiner indicated that "Applicant's [sic] argues [sic] that the directories of Frank are not used to locate the subpages, but Examiner disagrees, Fig. 5 shows a virtual address input to the directory, with an output that is used to access the cache." Appellants respectfully submit that FIG. 5 of *Frank* discusses "organization of the cache directory" (col. 12, line 44), but the cache directory is related to pages, not subpages or relocation blocks as in Appellants' claimed invention. Examples of discussions about *page* in FIG. 5 are "[t]he index in the set . . . identifies a page in the cache" (col. 12, lines 49-51); "the cache signal a missing page" (col. 12, line 52); "[t]wo important distributions cases are produced by referencing many pages . . ." (col. 12, lines 58-60); "[t]he use of SVA . . . to select a cache set produces good cache behavior for contiguous groups of pages" (col. 12, lines 61-63). Even though *Frank*'s FIG. 5 mentions subpages, *Frank* does not teach, suggest, or make obvious *using the cache directory to locate the subpages*. Some examples of mentioning subpages are "[d]escriptor.owner_limit" is propagated to other caches as follows: so long as all of the subpages of a descriptor are read-only copies" (col. 14, lines 59-61); "[w]hen a new cache becomes the owner of a subpage, it copies the value of descriptor.owner_limit from the old owner" (col. 14, lines 62-63).

Frank, even though discusses "finding the subpages" in response to a data access (col. 11, lines 59-63), Appellants respectfully submit that *Frank* does not teach, suggest, or make

obvious using the *cache directories* to locate the subpages that the Examiner has corresponded to the relocation blocks in the claimed invention.

3) Frank's subpage is patentably distinguished from Appellants' relocation block because Frank does not afford the flexibility that allows the subpages of a page to be at different locations including one or a plurality of memory systems

Based on the relocation table and its entries, embodiments of Appellants' invention allow relocation blocks of a memory page to be at different locations, and, as claimed in claim 1, the relocation blocks of a memory page are placed "at a plurality of locations including one or a plurality of memory systems." Further, upon a memory access, if the data intended for the memory access is not in physical memory, then Appellants' claim 1 recites "loading, in physical memory, one or a plurality of relocation blocks [at different locations including one or a plurality of memory systems] containing the data related to the memory access." In contrast, *Frank's* subpages do not afford such flexibility because "[e]ach page [that includes subpages] of SVA space is either *entirely* represented in the system or not represented at all" (col. 11, lines 27-29, emphasis added). In *Frank*, even if information only in a subpage is accessed, the *whole page* including the subpage must be loaded (col. 36, lines 51-55). Appellants' claim 1 is clearly patentably distinguished from this feature of *Frank* because, in Appellants' claim 1, only *one or a plurality of location blocks* or subpage, not the whole page, is loaded.

In paragraph 4 of the Final Office Action, the Examiner asserted "[t]he additional limitation that relocation blocks are placed in one or plural memory systems is disclosed since they are place [sic] in a memory system as cited hereinabove." In paragraph 5, the Examiner further asserted: "[t]he argument that Frank's pages must be represented by whole and not in part is not differentiated by the present claim language since one memory system reads on 'one or a plurality.'" Appellants respectfully submit that Appellants' claim 1 is further distinguished from *Frank* because within the page from which the blocks are divided, the blocks do not have to stay at the original location in the page. The blocks can be moved from one location to another location even in a page. Additionally, the relocation blocks do not have to stay within the boundary of a page from which the blocks are divided. Being identified by the entries of the relocation table, these relocation blocks can be at different memory systems that can be pointed to by the entries of the relocation table. As a result, these relocation blocks can be stored in *one or a plurality of memory systems*. In contrast,

within a boundary of a *Frank*'s page, a subpage cannot be moved from one location to another location, and every subpage of a page must stay within the page's boundary. Therefore, *Frank*, when being involved with a subpage, is involved with the *whole page*. That is why "[e]ach page [that includes subpages] of SVA space is either *entirely* represented in the system or *not represented at all*" (col. 11, lines 27-29, emphasis added).

On page 2 of the Advisory Opinion the Examiner indicated that the "Examiner disagrees with the remaining arguments also, since even though a page must be entirely represented, it may be distributed by way of different subpages in the directory caches. As indicated above, Appellants' relocation blocks, once being divided from a page, can be moved to locations identified by the table entries. An individual block does not have to stay with the boundary of the page from which the block is divided, may be moved to another location within the same page, in the same or different memory system, etc. As such, a relocation block of Appellants' invention is a unit independent from the page from which the block is divided. In contrast, *Frank*'s subpage must be *entirely represented by the page*."

Because claim 1 includes limitations patentably distinguished from *Frank*, claim 1 is patentable. Because independent claims 5 and 9 recite limitations corresponding to claim 1, claims 5 and 9 are patentable for at least the same reasons as claim 1.

Because dependent claims 2 – 4, 6 – 8, and 10 – 12 depend from claims 1, 5, and 9, respectively, claims 2 – 4, 6 – 8, and 10 – 12 are patentable for at least the same reasons as claims 1, 5, and 9, respectively. Claims 2 – 4, 6 – 8, and 10 – 12 are further patentable for their own limitations.

IX. CLAIMS INVOLVED IN THE APPEAL

A copy of the claims involved in the present appeal is attached hereto as Appendix A.

Application No.: 09/896,157

Docket No.: 10012379-1

I hereby certify that this correspondence is
being deposited with the United States
Postal Service as Express Mail,
Express Mail Label No.: EL442081774US
in an envelope addressed to: MS Appeal
Brief – Patents, Commissioner for Patents,
P.O. Box 1450, Alexandria, VA 22313-
1450.

Date of Deposit: Feb. 4, 2004

Typed Name: Jerome Dechant

Signature: Jerome Dechant

Respectfully Submitted,

By: Tuan Ngo

Tuan Ngo
Attorney or Appellants
Reg. No. 44,259

Date: 2/4/04

Telephone No. (408) 447-8133

APPENDIX A**Claims Involved in the Appeal of Application Serial No. 09/896,157**

1 1. (Previously Amended Once) A method for managing memory in a computer system,
2 comprising:
3 for at least one memory page,
4 dividing the page into a plurality of relocation blocks, and
5 placing the plurality of relocation blocks at a plurality of locations
6 including one or a plurality of memory systems; and
7 using a relocation table having a plurality of entries to locate the relocation
8 blocks at the plurality of locations;
9 wherein, upon a memory access,
10 using the relocation table to convert an address of the memory page
11 to a relocation address of a relocation block containing the
12 data intended for the memory access; and
13 if the data intended for the memory access is not in physical
14 memory, then loading, in physical memory, one or a
15 plurality of relocation blocks containing the data related to
16 the memory access.

1 2. (Original) The method of claim 1 further comprises the step of converting a virtual
2 address of the data to the address of the memory page.

1 3. (Original) The method of claim 1 further comprises the step of allocating the plurality
2 of relocation blocks corresponding to the memory page upon receiving the address
3 of the memory page.

1 4. (Original) The method of claim 3 further comprises the step of corresponding each
2 entry of the plurality of entries to a particular location of a relocation block.

1 5. (Previously Amended Once) A system for managing memory in a computer system,
2 comprising:
3 a plurality of relocation blocks located at a plurality of locations including
4 one or a plurality of memory systems; wherein a set of relocation
5 blocks is divided from a memory page;
6 a relocation table having a plurality of entries that is used to locate the
7 relocation blocks at the plurality of locations and to convert an
8 address of the memory page to a relocation address of a relocation
9 block containing the data intended for a memory access; and
10 if the data intended for the memory access is not in physical memory, then
11 loading, in physical memory, one or a plurality of relocation blocks
12 containing the data related to the memory access.

1 6. (Original) The system of claim 5 wherein the address of the memory page was
2 translated from a virtual address of the data.

1 7. (Original) The system of claim 5 further comprises means for allocating the plurality of
2 relocation blocks corresponding to the memory page upon receiving the address of
3 the memory page.

1 8. (Original) The system of claim 7 wherein each entry of the plurality of entries
2 corresponds to a particular location of a relocation block.

1 9. (Previously Amended Once) A computer-readable medium embodying instructions that
2 cause a computer to perform a method for managing memory in a computer
3 system, the method comprising the steps of:
4 for at least one memory page,
5 dividing the page into a plurality of relocation blocks, and
6 placing the plurality of relocation blocks at a plurality of locations
7 including one or a plurality of memory systems; and
8 using a relocation table having a plurality of entries to locate the relocation
9 blocks at the plurality of locations;
10 wherein, upon a memory access,
11 using the relocation table to convert an address of the memory page
12 to a relocation address of a relocation block containing the
13 data intended for the memory access; and
14 if the data intended for the memory access is not in physical
15 memory, then, loading, in physical memory, one or a
16 plurality relocation blocks containing the data related to the
17 memory access.

1 10. (Original) The computer-readable medium of claim 9 wherein the method further
2 comprises the step of converting a virtual address of the data to the address of the
3 memory page.

1 11. (Original) The computer-readable medium of claim 9 wherein the method further
2 comprises the step of allocating the plurality of relocation blocks

3 corresponding to the memory page upon receiving the address of the memory
4 page.

1 12. (Original) The computer-readable medium of claim 11 wherein the method further
2 comprises the step of corresponding each entry of the plurality of entries to a
3 particular location of a relocation block.